Ractors can't access global variables, but some special global variables should be accessed. There are several types.

**Proposal**

1. **Read-only global variables**
   
   ```ruby
   # process-local (readonly): other commandline parameters
   '-p' => $-p,
   '-l' => $-l,
   '-a' => $-a,
   
   # process-local (readonly): getpid
   '$$' => $$,
   ```

2. **Scope local variables**
   
   ```ruby
   # thread local: process result
   '$?' => $?,
   
   # scope local: match
   '$=' => $=.inspect,
   '$&' => $&,
   '$' => $,
   '$\n' => $'
   
   # scope local: last line
   '_.' => $_,
   
   # scope local: last backtrace
   '$@' => $@,
   '$!' => $!,
   ```

3. **Ractor local variables**
   
   ```ruby
   # ractor-local (derived from created ractor): debug
   '$DEBUG' => $DEBUG,
   '-d' => $-d,
   
   # ractor-local (derived from created ractor): verbose
   '$VERBOSE' => $VERBOSE,
   '-w' => $-w,
   '+W' => $-W,
   '-v' => $-v,
   
   # ractor local: stdin, stdout, stderr
   '$stdin' => stdin.inspect,
   '$stdout' => stdout.inspect,
   '$stderr' => stderr.inspect,
   ```
Implementation: https://github.com/ruby/ruby/pull/3670
I'll merge it soon.

**Discussion**

- only accessible from main ractor?
  - $0:
    - ARGV, ARGF, $.
  - only accessible from main ractor because they will be obsolete
    - $, $/; $\n  - So difficult: $" / $LOADED_FEATURES and $:. / $LOAD_PATH

**Associated revisions**

**Revision 99310e3e - 10/20/2020 06:38 AM - ko1 (Koichi Sasada)**
Some global variables can be accessed from ractors
Some global variables should be used from non-main Ractors.
[Bug #17268]

```ruby
# ractor-local (derived from created ractor): debug
'$_DEBUG' => $DEBUG,
'$_-d' => $-d,

# ractor-local (derived from created ractor): verbose
'$_VERBOSE' => $VERBOSE,
'$_-w' => $-w,
'$_-W' => $-W,
'$_-v' => $-v,

# process-local (readonly): other commandline parameters
'$_-p' => $-p,
'$_-l' => $-l,
'$_-a' => $-a,

# process-local (readonly): getpid
'$_$' => $$,

# thread local: process result
'$_?' => $?,

# scope local: match
'$_-' => $-.inspect,
'$_$' => $&,
'$_\' => $`,
'$_\'' => $'.,
'$_+$' => $+,
'$_1' => $1,

# scope local: last line
'$_' => $_,

# scope local: last backtrace
'$_$' => $0,
'$_!' => $!,

# ractor local: stdin, out, err
'$_stdin' => $stdin.inspect,
'$_stdout' => $stdout.inspect,
'$_stderr' => $stderr.inspect,
```

**History**

**#1 - 10/20/2020 01:49 AM - ko1 (Koichi Sasada)**
- Description updated

**#2 - 10/20/2020 06:39 AM - ko1 (Koichi Sasada)**
- Status changed from Open to Closed

Applied in changeset git|99310e3e3eb56fbc85bb45119285812eb959448d0c.
Some global variables can be accessed from ractors

Some global variables should be used from non-main Ractors.

[Bug #17268]

<table>
<thead>
<tr>
<th>Variable</th>
<th>Description</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$DEBUG</code></td>
<td>debug ractor-local</td>
<td><code>$DEBUG</code> =&gt; <code>$DEBUG</code></td>
</tr>
<tr>
<td><code>$-d</code></td>
<td>debug ractor-local</td>
<td><code>$-d</code> =&gt; <code>$-d</code></td>
</tr>
<tr>
<td><code>$VERBOSE</code></td>
<td>verbose ractor-local</td>
<td><code>$VERBOSE</code> =&gt; <code>$VERBOSE</code></td>
</tr>
<tr>
<td><code>$-v</code></td>
<td>verbose ractor-local</td>
<td><code>$-v</code> =&gt; <code>$-v</code></td>
</tr>
<tr>
<td><code>$-w</code></td>
<td>verbose ractor-local</td>
<td><code>$-w</code> =&gt; <code>$-w</code></td>
</tr>
<tr>
<td><code>$-W</code></td>
<td>verbose ractor-local</td>
<td><code>$-W</code> =&gt; <code>$-W</code></td>
</tr>
<tr>
<td><code>$-p</code></td>
<td>read-only process-local</td>
<td><code>$-p</code> =&gt; <code>$-p</code></td>
</tr>
<tr>
<td><code>$-l</code></td>
<td>read-only process-local</td>
<td><code>$-l</code> =&gt; <code>$-l</code></td>
</tr>
<tr>
<td><code>$-a</code></td>
<td>read-only process-local</td>
<td><code>$-a</code> =&gt; <code>$-a</code></td>
</tr>
<tr>
<td><code>$$$</code></td>
<td>get-pid thread-local</td>
<td><code>$$$</code> =&gt; <code>$$$</code></td>
</tr>
<tr>
<td><code>$?</code></td>
<td>process result scope-local</td>
<td><code>$?</code> =&gt; <code>$?</code></td>
</tr>
<tr>
<td><code>$~</code></td>
<td>scope-local match</td>
<td><code>$~</code> =&gt; <code>$~</code></td>
</tr>
<tr>
<td><code>$&amp;</code></td>
<td>scope-local last line</td>
<td><code>$&amp;</code> =&gt; <code>$&amp;</code></td>
</tr>
<tr>
<td><code>$\</code></td>
<td>scope-local last backtrace</td>
<td><code>$\</code> =&gt; <code>$\</code></td>
</tr>
<tr>
<td><code>$!</code></td>
<td>ractor-local stdin, out, err</td>
<td><code>$!</code> =&gt; <code>$!</code></td>
</tr>
<tr>
<td><code>$stdin</code></td>
<td>ractor-local stdin, out, err</td>
<td><code>$stdin</code> =&gt; <code>$stdin.inspect</code></td>
</tr>
<tr>
<td><code>$stdout</code></td>
<td>ractor-local stdin, out, err</td>
<td><code>$stdout</code> =&gt; <code>$stdout.inspect</code></td>
</tr>
<tr>
<td><code>$stderr</code></td>
<td>ractor-local stdin, out, err</td>
<td><code>$stderr</code> =&gt; <code>$stderr.inspect</code></td>
</tr>
</tbody>
</table>

#3 - 10/22/2020 02:56 PM - Dan0042 (Daniel DeLorme)

Why can't ractors access global variables? Shouldn't it be like constants, where a frozen/shareable object can be accessed?

```ruby
$x = [1, 2, 3]
Ractor.new { $x }.take # I understand that this can't work
$x = true
Ractor.new { $x }.take # but why is this not ok?
```

#4 - 10/22/2020 07:31 PM - Eregon (Benoit Daloze)

I think this issue is mostly about "special" global variables, not so much about "regular" global variables.

I wonder if "regular" global variables should be Ractor-local, or behave like constants. Since most of the above became Ractor-local (instead of per-Process before) or a narrower scope, I think Ractor-local would make most sense. The only special variables that are still process-local are read-only, which means it's the same as if they were Ractor-local read-only.

#5 - 10/22/2020 09:18 PM - Dan0042 (Daniel DeLorme)

Eregon (Benoit Daloze) wrote in #note-4:

I wonder if "regular" global variables should be Ractor-local, or behave like constants.

I wouldn't mind either way. Although if Ractor-local I would hope they are derived from created ractor, like $DEBUG and $VERBOSE.

#6 - 10/26/2020 04:36 PM - ko1 (Koichi Sasada)

04/24/2021
The proposal is accepted.

I wonder if "regular" global variables should be Ractor-local, or behave like constants.

This is rejected at first by Matz (my original proposal is making all $gvars are ractor-local), to avoid confusion.